updated manifest #1 through updated manifest #M corresponding, respectively, to the updated components **306**. Although the installation instruction sets **304**, updated components **306**, and updated manifests **308** are illustrated together in service package **302**, alternatively any one or more of the installation instruction sets **304**, updated components **306**, and updated manifests **308** may be stored at different locations (either remote or local to computing device **201**).

[**0038**] The installation instruction sets **304** identify the components **306** that are updates for operating system **202** (e.g., including the versions of components **306**). During an update process, assuming that the new components **306** have not already been installed as one of components **206**, the updated components **306** and corresponding manifests **308** replace or supplement the corresponding components **206**. By way of example and not limitation, component **306** is a newer version of component **206**. In this situation, component **306** and manifest **308** would replace component **206** and manifest **208**, respectively, in operating system **202**. Component **306** may replace component **206** by simply overwriting one or more of the files in component **206** by one or more of the files in updated component **306** (the overwritten files may optionally be saved elsewhere before being overwritten). In a similar example, if manifest **308** indicates that component **306** lacks a file that currently exists in component **206** (e.g., as indicated by reviewing manifest **208**), the file is deleted from the computer-readable medium storing operating system **202** during the update process.

[**0039**] In another embodiment, the files in updated component **306** may simply be a reference to other files. For example, to minimize the size of service package **302** and to ensure that the latest updated components **306** are installed, the updated components **306** may include hyperlinks to the actual files to be installed. The component installer **210** downloads the files associated with the updated components **306** during the update process.

[**0040**] The operating system **202** may be updated for any of a wide variety of reasons. For example, bug fixes to certain files of certain components may be available, new functionality (e.g., replacement or additional files) for component **206** may be available, or new components **306** may be available.

[**0041**] Additionally, a new component **306** may be installed as part of the operating system **202** in addition to a previous component **206** rather than in place of the previous component **206**. In this example, the manifest **208** is modified to indicate that both the new component **306** and the previous component **206** are installed. This allows different applications to use whichever version of the components **206**, **306** they prefer (or are programmed to use). Those skilled in the art will note that in some embodiments, the instruction sets **304** are part of the manifests **308** for each component. In this manner, the component installer accesses each manifest **308** to determine how to install the component **306** associated therewith.

[**0042**] Operation of the Component Installer

[**0043**] Referring next to **FIG. 4**, an exemplary flow chart illustrates operation of the component installer. The process of **FIG. 4** is implemented by an application program such as component installer **210** that may be executed on a computer such as computing device **201** or alternatively on another computer coupled to the computer. The component installer may be part of the software product (e.g., operating system **202**) being updated or, alternatively, a separate application. The update process of **FIG. 4** may be performed in software, hardware, firmware, or a combination thereof.

[**0044**] The update process includes receiving the service package, determining the state associated with the component at **402**, selecting one of the instruction sets based on the determined state at **404**, and applying one or more of the files to the component in accordance with the selected instruction set at **406**. Applying the files includes copying each of the files from the service package to the computer-readable medium storing the component and integrating the copied files with the component. Alternatively or in addition, applying the files includes performing a file action and/or a data action. Performing the file action includes copying, deleting, and replacing files. Performing the data action includes updating a system setting such as a registry entry.

[**0045**] The update process further includes updating the manifest for the component with data related to applying the one or more of the files and storing the updated manifest for the component with the component at **408**. The stored data comprises at least one of the following: a version number, an update time, a description of the service package, a description of the changes made to the component, the current lifecycle phase of the software product, and a storage location of the component. The update process further includes receiving a request from a user for the data in the manifest at **410**, querying the manifest in response to the received request to generate query results at **412**, and providing the query results to the user at **414**.

[**0046**] The update process also resolves dependencies between the updated components and the existing components in the software product. Dependency resolution includes accessing the selected instruction set to identify one or more additional components dependent on or by the updated component. As a result of the dependency resolution, the component installer may request and receive additional components as needed.

[**0047**] One or more computer-readable media have computer-executable instructions for performing the method illustrated in **FIG. 4**.

[**0048**] Creating the Service Package

[**0049**] Referring next to **FIG. 5**, an exemplary flow chart illustrates creation of a service package. An original equipment manufacturer (OEM), vendor, system administrator, independent software vendor, or other user creates the service package for distribution on a computer-readable medium or over a network. The method includes selecting one or more files for association with the software product at **502** and storing the selected files on a computer-readable medium **504**. The method also stores a plurality of installation scripts on the computer-readable medium at **506**. Each of the installation scripts corresponds to one of the states of the components to be updated. The installation scripts are executed based on the state of the component to apply the stored, selected files to the software product. One or more computer-readable media have computer-executable instructions for performing the method illustrated in **FIG. 5**.